Journal of Discrete Algorithms 10 (2012) 165–180

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Running time analysis of Ant Colony Optimization for shortest path problems ☆

Dirk Sudholt [a,*], Christian Thyssen [b,1]

[a] CERCIA, University of Birmingham, Birmingham B15 2TT, UK
[b] Technische Universität Dortmund, 44221 Dortmund, Germany

ARTICLE INFO

Article history:
Received 24 September 2010
Received in revised form 6 June 2011
Accepted 6 June 2011
Available online 12 June 2011

Keywords:
Ant Colony Optimization
Combinatorial optimization
Metaheuristics
Running time analysis
Shortest path problems

ABSTRACT

Ant Colony Optimization (ACO) is a modern and very popular optimization paradigm inspired by the ability of ant colonies to find shortest paths between their nest and a food source. Despite its popularity, the theory of ACO is still in its infancy and a solid theoretical foundation is needed. We present bounds on the running time of different ACO systems for shortest path problems. First, we improve previous results by Attiratanasunthron and Fakcharoenphol [Information Processing Letters 105 (3) (2008) 88–92] for single-destination shortest paths and extend their results from DAGs to arbitrary directed graphs. Our upper bound is asymptotically tight for large evaporation factors, holds with high probability, and transfers to the all-pairs shortest paths problem. There, a simple mechanism for exchanging information between ants with different destinations yields a significant improvement. A comparison with evolutionary and genetic approaches indicates that ACO is among the best known metaheuristics for the all-pairs shortest paths problem.

1. Introduction

Ant Colony Optimization (ACO) is a rapidly growing field with many successful applications to problems from combinatorial optimization. It is inspired by the foraging behavior of real ants, which enables an ant colony to find shortest paths between its nest and a food source. Ants communicate by placing pheromone on the ground while searching the environment for food. Other ants are attracted by pheromone trails and therefore tend to follow previous ants. In case foraging ants discover different paths between the nest and a food source, a short path typically gets invested with pheromone more quickly than a longer path. The more ants take the short path, the more pheromone is deposited, until almost all ants follow the short path.

The communication mechanism of real ants has been transferred to many optimization problems such as the TSP [14], routing problems [6,15], and many other combinatorial problems, see the book by Dorigo and Stützle [16]. These algorithms often perform extremely well in practice and they often produce better results than deterministic algorithms. Moreover, they are applicable even in settings where the optimization problem is not well enough understood to design custom-tailored algorithms. In an extreme case, the problem is given as a black box and evaluations of candidate solutions are the only way to gather information about the problem. This setting is known as black-box optimization [17].

☆ A preliminary version of this article has been presented at SLS '09 (Horoba and Sudholt, 2009 [23]).

* Corresponding author.
E-mail addresses: d.sudholt@cs.bham.ac.uk (D. Sudholt), christian.thyssen@ls2.cs.tu-dortmund.de (C. Thyssen).

[1] Horoba is the former name of the second author.

1570-8667/$ – see front matter
doi:10.1016/j.jda.2011.06.002

Despite a plethora of applications, the theoretical knowledge on ACO is still very limited. It is widely agreed that a solid theoretical foundation is needed [13,20,45]. First theoretical investigations concerned convergence proofs [19] and simplified models of ACO algorithms [29]. In 2006 the first rigorous investigations of the running time of ACO algorithms were presented independently by Gutjahr [21] and Neumann and Witt [35,37] for the optimization of simple pseudo-Boolean functions. The latter authors presented an algorithm called 1-ANT. This algorithm memorizes the best solution found so far. In each iteration a new solution is constructed and the pheromones are updated in case another solution with at least the same quality is found. In other words, every new best-so-far solution is rewarded only once. The strength of the pheromone update is usually controlled by a parameter $0 < \rho \leqslant 1$ known as *evaporation factor*. The smaller the value of ρ, the larger the impact of previous pheromones is, compared to the impact of newly placed pheromones.

Investigations of the 1-ANT [11,37] have shown that if the evaporation factor is set too small the algorithm stagnates on even very simple problems and the expected time until an optimum is found is exponential. Other algorithms, variants of the MAX-MIN Ant System (MMAS) [40], in every iteration reinforce the best solution found so far. This update strategy is known as *best-so-far update*; it avoids the problem of stagnation and leads to efficient running times on various problems [22,28,32]. Moreover, reinforcing the best solution created in the current iteration, known as *iteration-best update*, can lead to efficient running times if the evaporation factor is small enough [33,42].

Neumann, Sudholt, and Witt [31] investigated the effect of hybridizing ACO with local search. They demonstrated for artificially constructed problems that the use of local search can turn an exponential runtime into a polynomial one, with high probability. On another constructed function, the effect is reversed. Regarding ACO for combinatorial problems, Neumann and Witt [36] presented an analysis for minimum spanning trees. They proved polynomial upper bounds for different mechanisms of ants constructing spanning trees. They also addressed the impact of using heuristic information to guide the construction process. Zhou [46] presented an analysis of ACO where he proved polynomial running time bounds on simple instances of the TSP. This study was later significantly extended by Kötzing, Neumann, Röglin, and Witt [27]. Furthermore, Attiratanasunthron and Fakcharoenphol [1] presented a running time analysis of ACO algorithms on a shortest path problem, the single-destination shortest path problem (SDSP) on directed acyclic graphs (DAGs). Their algorithm n-ANT is inspired both by the 1-ANT [35] and the AntNet algorithm [6]. To our knowledge, this is the first and only rigorous running time analysis for ACO on a shortest path problem. This is surprising as shortest path problems crucially inspired the development of ACO.

The aim of this work is to develop deeper insights into the working principles of ACO and on the performance of ACO algorithms on shortest path problems. We choose shortest path problems not only because of their relation to natural ant colonies. These problems are well-understood and have a clear structure, which makes them an excellent starting point for theoretical investigations. Besides, shortest path problems are a fundamental topic in computer science and algorithmic research on these problems is still an active field [2,4,38,44].

1.1. Previous work

Shortest paths have already been investigated in the context of other metaheuristics. We present a brief survey of these results. Note that in the theory of randomized search heuristics, it is common practice to consider the number of evaluations of the objective function, i.e., the number of path length evaluations. (See Section 2.2 how this relates to the usual notion of time complexity on random access machines.) This makes sense as often function evaluations are among the computationally most expensive operations in such a heuristic. Unless mentioned otherwise, the number of function evaluations in the reviewed results has the same order of growth as the number of iterations.

Scharnow, Tinnefeld, and Wegener [39] presented an analysis of a simple evolutionary algorithm for the single-source shortest path problem (SSSP). Note that the problem SSSP is obtained from SDSP by inverting the direction of all edges of the graph. Their algorithm, called $(1+1)$ EA, maintains a current solution v for the problem, encoded by a shortest path tree. In each iteration a mutation operator is used to create a new solution v' with random changes in the shortest path tree. If all shortest paths in v' are not worse than all shortest paths in v then v' replaces v. The authors prove that this simple algorithm solves the SSSP on every n-vertex graph in an expected number of $O(n^3)$ iterations. These results were later refined by Doerr, Happ, and Klein [7] who gave a tight bound of $O(n^2\ell^*)$ iterations, where $\ell^* := \max\{\ell, \ln n\}$ and ℓ is, loosely speaking, the maximum number of edges on any shortest path. The comparison of two solutions in the $(1+1)$ EA is based on a multiobjective formulation of a single-objective problem as all paths in one solution are compared with their respective counterparts in the other solution. Baswana et al. [3] considered a variant of the $(1+1)$ EA using a single-objective function and proved an upper bound of $O(n^3(\log n + \log w_{\max}))$ where $w_{\max}$ denotes the largest weight in the graph. This result was recently improved by Doerr and Johannsen [9] towards a bound of $2emn(\ln n + \ln w_{\max})$, m being the number of edges, for a mutation operator that focuses on edges instead of vertices. The same modification improves the result from Doerr, Happ, and Klein [7] towards a bound of $O(m\ell^*)$.

In [8] Doerr, Happ, and Klein investigated a genetic algorithm, simply called GA, for the all-pairs shortest path problem (APSP) that used both mutation and crossover operators to create new solutions. A solution is represented by a "population" of paths with different start and end points. The crossover operator concatenates edges taken from different paths. The authors proved that the use of crossover leads to a speed-up compared to mutation-based evolutionary algorithms; a fact that previously was only proven for artificial problems [25] and a simple graph coloring problem [18,41]. More precise, the introduction of crossover decreases the expected number of iterations from $\Theta(n^4)$ to $O(n^{3.5}\log^{0.5} n)$. The latter upper bound

Table 1
Overview on the best known running time bounds on graphs with n vertices, m edges, maximum degree Δ, maximum number of edges ℓ on any shortest path, and $\ell^* := \max\{\ell, \ln n\}$. The bounds for ACO algorithms are stated with their dependence on the parameter ρ. The rightmost column contains the number of path length evaluations per iteration. For all ACO algorithms the best feasible values for the parameter $\tau_{\min}$ have been chosen. The bound for $\text{MMAS}_{\text{APSP}}$ with interaction holds for $\rho \leqslant (1 - p_{\text{int}})/(12\Delta \log n)$; it simplifies to $O(n \log n + \Delta \log^3 n)$ for optimal ρ and $p_{\text{int}} = 1/2$.

Algorithm	Problem	Iterations	Evaluations
n-ANT [1]	SDSP on DAGs	$O\left(\frac{m\Delta\ell \log(\Delta\ell)}{\rho}\right)$	n
$\text{MMAS}_{\text{SDSP}}$	SDSP	$O\left(\Delta\ell\ell^* + \frac{\ell}{\rho}\right)$	n
$\text{MMAS}_{\text{SDSP}}$	SDSP on $G^{\text{lb}}_{n,\ell}$	$\Omega\left(\Delta\ell^2 + \frac{\ell}{\rho \log(1/\rho)}\right)$	n
$\text{MMAS}_{\text{SDSP}}$ + adaptive $\tau_{\min}$	SDSP	$O\left(\ell m + \frac{n}{\rho}\right)$	n
$(1+1)$ EA, vertex-based [7]	SSSP	$\Theta(n^2\ell^*)$	1
$(1+1)$ EA, edge-based [9]	SSSP	$\Theta(m\ell^*)$	1
$\text{MMAS}_{\text{APSP}}$	APSP	$O\left(\Delta\ell\ell^* + \frac{\ell}{\rho}\right)$	n^2
$\text{MMAS}_{\text{APSP}}$ + interaction	APSP	$O\left(n \log n + \frac{\log(\ell)\log(\Delta\ell)}{\rho}\right)$	n^2
GA [12]	APSP	$\Theta(n^{3.25} \log^{0.25} n)$	1
GA + feasible selection [10]	APSP	$\Theta(n^3 \log n)$	1

was improved by Doerr and Theile [12] to $O(n^{3.25} \log^{0.25} n)$ iterations. An example is given where the improved bound is tight. With a slightly more sophisticated parent selection mechanism that guarantees that the created offspring are feasible, the last result can be improved to $O(n^3 \log n)$ as recently shown by Doerr, Johannsen, Kötzing, Neumann, and Theile [10].

Finally, Horoba (http://dx.doi.org/10.1162/EVCO_a_00014) investigated an NP-hard multiobjective shortest path problem where each edge is labelled with a weight vector. He proved that an evolutionary multiobjective algorithm represents a fully polynomial-time randomized approximation scheme for the problem. Neumann and Theile [34] presented an extension for the multiobjective APSP that has a significantly lower expected runtime than running n evolutionary algorithms for multiobjective SPSP problems independently – when dealing with up to three weights per edge.

In the mentioned work [1] on ACO for shortest paths, Attiratanasunthron and Fakcharoenphol prove an upper bound of $O(m\Delta\ell \log(\Delta\ell)/\rho)$ for every directed acyclic graph with m edges, maximum degree Δ and maximum number of edges ℓ on any shortest path. The evaporation factor ρ is a parameter of the algorithm and can be chosen arbitrarily, subject to $0 < \rho \leqslant 1$. Their analysis follows the analysis of the Bellman–Ford algorithm [5]. We will review their algorithm and their analysis in more detail.

In the meantime the authors also presented results for a stochastic shortest path problem where edge weights are subjected to added noise [24] and the problem is to find the real shortest paths despite the noise. It is shown that in a setting with independent gamma-distributed noise there exist graphs where the expected time until $\text{MMAS}_{\text{SDSP}}$ finds a good approximation of the real shortest paths is exponential.

We summarize the number of iterations needed to find shortest paths and the number of function evaluations performed in a single iteration. Table 1 gives an overview on the best known bounds for different heuristics in the case of single-objective problems, including bounds that will be proven in this article.

For the sake of completeness, we also mention results for problem-specific algorithms, keeping in mind that we cannot hope a general-purpose metaheuristic like ACO to outperform tailored algorithms. It is well known that Dijkstra's algorithm can solve SDSP for graphs with n vertices and m edges in time $O(m + n \log n)$ and APSP in time $O(nm + n^2 \log n)$ [5]. The best currently known general bound for the APSP is $O(n^3(\log^3 \log n)/\log^2 n)$ [4]. All ACO algorithms discussed in this work can be regarded as path-comparison-based algorithms as defined by Karger, Koller, and Phillips [26]. The latter authors also proved a lower bound of $\Omega(mn)$ for every deterministic or randomized path-comparison-based algorithm for the APSP. This bound even holds for generalized path weight functions.

1.2. Outline

The remainder of this article is structured as follows. In Section 2 we define the ACO algorithm $\text{MMAS}_{\text{SDSP}}$ for the SDSP that differs from the n-ANT [1] in two essential ways. Using our modified algorithm we are able to obtain significantly improved running time bounds (see Table 1 and Section 3) and to generalize previous results for DAGs to graphs with cycles. A corresponding lower bound shows that our upper bounds are asymptotically tight if the evaporation factor ρ is not too small. In Section 4 we transfer these results to a generalized ant system $\text{MMAS}_{\text{APSP}}$ for the APSP where ants with different destinations move independently. The main result concerns a modification of $\text{MMAS}_{\text{APSP}}$ where ants temporarily follow foreign pheromone traces. We prove that, surprisingly, this simple mechanism leads to a significant speed-up. We conclude in Section 5 with remarks on generalizations and future work.

2. ACO algorithms for shortest paths

2.1. Preliminaries

We consider shortest path problems on weighted directed graphs $G = (V, E, w)$ where $w(e)$ denotes the weight of edge e. The number of vertices is always denoted by n. We define a *path* of length ℓ from u to v as a sequence of

Algorithm 1 Path construction from u to v for $\text{MMAS}_{\text{SDSP}}$

1: Initialize $i \leftarrow 0$, $p_0 \leftarrow u$, and $V_1 \leftarrow \{p \in V \setminus \{p_0\} \mid (p_0, p) \in E\}$
2: **while** $p_i \neq v$ and $V_{i+1} \neq \emptyset$ **do**
3: $\quad i \leftarrow i + 1$
4: $\quad$ Choose $p_i \in V_i$ with probability $\tau((p_{i-1}, p_i)) / \sum_{p \in V_i} \tau((p_{i-1}, p))$
5: $\quad V_{i+1} \leftarrow \{p \in V \setminus \{p_0, \ldots, p_i\} \mid (p_i, p) \in E\}$
6: **end while**
7: **return** $(p_0, \ldots, p_i)$

vertices $(v_0, \ldots, v_\ell)$ where $v_0 = u$, $v_\ell = v$, and $(v_{i-1}, v_i) \in E$ for all i with $1 \leqslant i \leqslant \ell$. For convenience, we also refer to the corresponding sequence of edges as path. Let $\deg(u)$ denote the out-degree of a vertex u and $\Delta(G)$ denote the maximum out-degree of any vertex $u \in V$. Let $\ell(G, v) := \max_u\{\#\text{edges on } p \mid p \text{ is a shortest path from } u \text{ to } v\}$ and $\ell(G) := \max_v \ell(G, v)$. For undirected non-weighted graphs $\ell(G, v)$ is called *eccentricity* of v and $\ell(G)$ *diameter* of G.

For the single-destination shortest path problem (SDSP) we are looking for shortest paths from every vertex to a specified destination vertex. The length $w(p)$ of a path p is defined as the sum of weights for all edges in p if the path ends with the destination vertex. If the path does not reach the destination, we define $w(p) := \infty$. Our results are restricted (either explicitly or implicitly) to graphs where all cycles have a strictly positive total weight. With negative-length cycles one can find arbitrarily short paths. In addition, the problem of computing a shortest *simple* path is NP-hard if arbitrary negative weights are allowed.

2.2. ACO algorithms

Attiratanasunthron and Fakcharoenphol [1] present the ACO algorithm n-ANT for the SDSP. Their algorithm is inspired by the 1-ANT [35] and the AntNet routing algorithm [6]. From every vertex $u \in V$ an ant a_u starts heading for the destination. The path is chosen by performing a random walk through the graph according to pheromones on the edges. Ant a_u memorizes the best path it has found from u to the destination so far. If it has found a path that is at least as good as the previous best-so-far path, a pheromone update takes place and the new path is reinforced. The authors use a purely local update rule: each ant a_u is responsible for updating the edges leaving its start vertex u. If the new path is worse, the pheromones on the edges leaving u remain unchanged.

As the authors only consider acyclic graphs, the n-ANT is not supposed to deal with cycles. In particular, in [1] the authors state that in graphs with cycles their path construction procedure might take exponential time. Therefore, we modify their algorithm in that we only allow ants to construct simple paths, i.e., an ant cannot visit a vertex more than once. The choice which edge to take next is made among all edges leading to unvisited vertices. This restriction bears the risk that the ant does not reach the destination. Recall that in this case the length of the path found is defined as $w(p) = \infty$. Due to the local pheromone update it is guaranteed that still one outgoing edge is rewarded for every vertex u with $\deg(u) \geqslant 1$ and $u \neq n$. The construction procedure is described in Algorithm 1.

We call our algorithm $\text{MMAS}_{\text{SDSP}}$ as we use the best-so-far update rule from the algorithm MMAS in [32] instead of the update rule used by the 1-ANT. The difference is that we always perform a pheromone update with the current best-so-far path, either with a new path or with the previous best-so-far path in case the new path is worse.

The update scheme is essentially taken over from [1]. We initialize the pheromones $\tau : E \to \mathbb{R}_0^+$ such that all edges leaving some vertex u receive the same amount of pheromone: if $e = (u, \cdot)$ then $\tau(e) = 1/\deg(u)$. If e is the only edge leaving u, we keep $\tau(e) = 1$ fixed. This means that vertices with a single outgoing edge are traversed in the only possible way; these vertices may therefore be disregarded when proving upper bounds on the running time. In case u has more than one outgoing edge, the pheromone for $e = (u, v)$ is computed as follows. Let p_u^* denote the best path from u found so far. Initially, we set p_u^* to an empty path, such that $w(p_u^*) = \infty$ by definition of w.

A common feature of ACO algorithms is that pheromone laid by previous ants evaporates over time. During a pheromone update usually a ρ-fraction of the old pheromone evaporates and hence a $(1 - \rho)$-fraction remains. Then new pheromone is added for the edges that shall be rewarded. Another common practice in ACO is to maintain fixed upper and lower bounds on the pheromones [1,40]. We keep pheromones within an interval $[\tau_{\min}, \tau_{\max}]$ where $\tau_{\min} > 0$ is a parameter of the algorithm and, in the remainder of this paper, $\tau_{\max} = 1 - \tau_{\min}$. These pheromone borders ensure that the algorithm can create any fixed solution with a positive probability. Hence it is guaranteed that the expected optimization time is finite. The precise formula for an update of the pheromones is as follows.

$$\tau(e) \leftarrow \begin{cases} \min\{(1-\rho)\cdot\tau(e) + \rho,\ \tau_{\max}\} & \text{if } e = (u, v) \in p_u^*, \\ \max\{(1-\rho)\cdot\tau(e),\ \tau_{\min}\} & \text{if } e = (u, v) \notin p_u^*. \end{cases} \tag{1}$$

We remark that the pheromone update formula equals the one used for the n-ANT [1], but the invocation of the update procedure is different in the n-ANT and $\text{MMAS}_{\text{SDSP}}$.

The complete algorithm $\text{MMAS}_{\text{SDSP}}$ is shown in Algorithm 2. We are interested in the *optimization time* or *running time* of $\text{MMAS}_{\text{SDSP}}$, defined as the number of iterations until shortest paths from all nodes $1, \ldots, n$ to the node n have been found. As mentioned before, a common performance measure for the analysis of metaheuristics is the number of function evaluations. Note that in one iteration of $\text{MMAS}_{\text{SDSP}}$ we have n ants constructing n solutions and performing n function

Algorithm 2 $\text{MMAS}_{\text{SDSP}}$

1: Initialize pheromones τ and best-so-far paths $p_1^*, \ldots, p_n^*$
2: **loop**
3: **for** $u = 1$ to n **do**
4: Construct a simple path $p_u = (p_{u,0}, \ldots, p_{u,\ell_u})$ from u to n w.r.t. τ
5: **if** $w(p_u) \leqslant w(p_u^*)$ **then** $p_u^* \leftarrow p_u$ **end if**
6: **end for**
7: Update pheromones τ w.r.t. $p_1^*, \ldots, p_n^*$
8: **end loop**

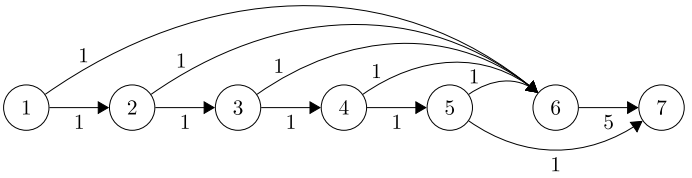

Fig. 1. Example graph for $n = 7$.

evaluations in parallel. Hence, the number of function evaluations is by a factor n larger than the number of iterations. We focus on the number of iterations in the following.

The obvious question arises how the optimization time relates to the computation time of the ant system $\text{MMAS}_{\text{SDSP}}$ on a random access machine. If we consider a naive implementation using adjacency lists, the running time for the initialization of the ant system is $O(m+n)$ because m pheromone values and n best-so-far paths have to be prepared. In the worst case, the running time for each iteration is $O(mn)$ because n ants construct a *simple* path each and the random construction of a simple path demands a running time of $O(m)$. Note that the latter estimation is pessimistic for many input instances and many vertices. More sophisticated implementations might reduce the total computation time per iteration in many cases. In addition, the use of several machines can help because each ant constructs its path independently from the other ants. Hence, using n machines to simulate all n ants in parallel improves the computation time per iteration by a factor of $1/n$.

2.3. Why a colony is needed

Before analyzing $\text{MMAS}_{\text{SDSP}}$ we motivate why it is essential to have ants starting from every vertex, even when we are only interested in a shortest path from a single source to a single destination and even when considering simple DAGs. Imagine a variant $\text{MMAS}_{\text{SPSP}}$ (SPSP for single-pair shortest paths) where one or multiple ants start from a single vertex, searching for the destination. Consider the following graph $G = (V, E, w)$ sketched in Fig. 1. It contains a single heavy edge $(n-1, n)$ with weight $n-2$ and light edges $(u, n-1)$ for $u \leqslant n-2$, $(u, u+1)$ for $u \leqslant n-3$, and $(n-2, n)$ of weight 1, each.

On each vertex $u \leqslant n-2$ an ant has to decide whether to move to $n-1$ or to proceed on the shortest path $1, 2, \ldots, n-2, n$. As all edges initially have equal pheromone, the probability that an ant follows the shortest path to vertex $n/2$ is $2^{-n/2+1}$. Assume the ant leaves the shortest path before reaching $n/2$. As the length of a path containing ℓ edges and traversing $n-1$ is $\ell + n - 3$, no further path with a larger number of edges will be accepted in the following, except for the optimal path traversing $1, 2, \ldots, n-2, n$. This implies that the pheromones for edges leaving the vertices $n/2, \ldots, n-2$ will always remain equal, unless an ant finds the optimum. The probability of finding the optimum is $2^{-n/2+1}$, hence taking the union bound over 2^{cn} steps for some small constant $c > 0$, the optimization time is at least 2^{cn} with probability $1 - 2^{-\Omega(n)}$. Note that this also holds in case polynomially many ants search for the destination in parallel in one iteration. By increasing the weight of the heavy edges from $n-2$ to $n-2+k$, $k \in \mathbb{N}$, we see that the ant system does not even compute a path approximating the weight of a shortest path by a factor of less than $(n-1+k)/(n-2) = 1 + (k+1)/(n-2)$.

Also using edge weights as heuristic information does not help. Many ACO algorithms use both pheromones and a heuristic function to guide the solution construction [16]. A common approach is to adjust the probability in line 4 of Algorithm 1 according to

$$\frac{[\tau((p_{i-1}, p_i))]^\alpha \cdot [\eta((p_{i-1}, p_i))]^\beta}{\sum_{p \in V_i} [\tau((p_{i-1}, p))]^\alpha \cdot [\eta((p_{i-1}, p))]^\beta}$$

where $\eta(e) = 1/w(e)$ denotes the heuristic information and the parameters $\alpha \in \mathbb{R}_0^+$ and $\beta \in \mathbb{R}_0^+$ control the relative importance of the pheromone versus the heuristic information. However, from a vertex $n/2 \leqslant u \leqslant n-2$ both outgoing edges have the same weight and the same pheromone, with high probability, hence they look the same for every ant. This example shows that heuristic information might be useless for some problem instances. In addition, decreasing the weight of the light edges to vertex $n-1$ and increasing the weight of the heavy edge to vertex n shows that heuristic information can even be misleading.

3. Single-destination shortest path problem

3.1. A first upper bound for $\text{MMAS}_{\text{SDSP}}$

When ants start from different vertices, ants starting close to the destination have a good chance of finding a shortest path. The pheromones deposited on the outgoing edges of a vertex v can then be used to guide different ants traversing v. This way, the shortest path for v can be extended towards a longer shortest path that contains v. In other words, shortest paths are subsequently propagated throughout the graph. This is the basic idea of the analysis by Attiratanasunthron and Fakcharoenphol [1], which is in turn based on the analysis of the Bellman–Ford algorithm [5]. The results in [1] are limited to directed acyclic graphs. We start with these graphs and extend the results to directed graphs with cycles. Thereby, we improve upon the upper bounds from [1] and present lower bounds which are tight with our best upper bounds for a broad range of evaporation factors ρ.

The following lemma will be used to estimate the probability of an ant choosing a specific edge.

Lemma 1. *For every vertex u, $u \neq n$, with $\deg(u) > 1$ always*

$$1 \leqslant \sum_{e=(u,\cdot)\in E} \tau(e) \leqslant 1 + \deg(u)\tau_{\min}.$$

Proof. The first inequality has already been proven in [1]. Initially the sum of pheromones equals 1. Assume for an induction that $\sum \tau(e) \geqslant 1$. If the pheromones are not capped by pheromone borders, we have $(1-\rho)\sum \tau(e) + \rho \geqslant 1$ as new sum. In case a pheromone drops below $\tau_{\min}$, setting the pheromone to $\tau_{\min}$ can only increase the sum. If at least one pheromone is capped at the upper border $\tau_{\max}$ then the sum of pheromones is at least $\tau_{\min} + \tau_{\max} = 1$ as $\deg(u) > 1$.

For the second inequality observe that the sum of pheromones can only increase due to the lower pheromone border as $(1-\rho)\sum \tau(e) + \rho \leqslant \sum \tau(e)$ follows from $\sum \tau(e) \geqslant 1$. Consider an edge e with $(1-\rho)\tau(e) < \tau_{\min}$. Compared to this value, the pheromone increases by at most $\tau_{\min} \cdot \rho$ when setting the pheromone to $\tau_{\min}$. If currently $\sum \tau(e) \leqslant 1 + \deg(u)\tau_{\min}$ then the sum of the next pheromone values is at most $(1-\rho)(1+\deg(u)\tau_{\min}) + \rho + \deg(u)\tau_{\min} \cdot \rho = 1 + \deg(u)\tau_{\min}$. Hence, the second inequality follows by induction. □

As an immediate consequence, we obtain the following direct relation between pheromones and probabilities for the ant a_u, i.e., the ant starting at u, of choosing an edge $(u,\cdot)$ in case $\tau_{\min} \leqslant 1/\deg(u)$. The last condition makes sense as $\tau_{\min}$ should be chosen below the initial pheromone value of $1/\deg(u)$.

Corollary 2. *If $\tau_{\min} \leqslant 1/\deg(u)$ for every edge $e = (u,\cdot)$, $u \neq n$,*

$$\tau(e)/2 \leqslant \text{Prob}(\textit{ant } a_u \textit{ chooses edge } e) \leqslant \tau(e).$$

The lower bound also holds for every other ant leaving vertex u and every edge $e = (u,v)$ unless v has already been traversed by the ant. The upper bound also holds for every other ant and every edge $e = (u,\cdot)$ if it has not traversed a successor of u before arriving at u.

The penultimate statement holds as the probability of choosing an edge $e = (u,v)$ to an unvisited successor v increases if other successors of u have been visited before. In particular, we always have $\tau_{\min}/2$ as lower bound on the probability of choosing any specific outgoing edge. This is an improvement to Lemma 1 in [1]. We remark that using the improved lemma in [1], the running time bounds for the algorithm n-ANT can be divided by m/n, where m is the number of edges.

The subsequent corollary bounds the probability that any ant proceeding on any shortest path follows an incorrect edge leaving the last vertex of the path. In the following, we call an edge (u,v) *correct* if it belongs to a shortest path from u to n and *incorrect* otherwise.

Corollary 3. *Consider a weighted graph $G = (V, E, w)$ where all cycles have strictly positive weight. Let $u, v \in V \setminus \{n\}$. Assume that the ant a_u has followed a shortest path from u to v where v is a vertex on a shortest path from u to n. If $\tau_{\min} \leqslant 1/\deg(u)$ then*

$$\text{Prob}\big(\textit{ant } a_u \textit{ chooses some incorrect edge } (v,\cdot)\big) \leqslant \tau_{\text{inc}}$$

where τ_{inc} denotes the sum of pheromone on incorrect edges $(v,\cdot)$.

Proof. The claim immediately follows from Corollary 2 when the graph is acyclic. When considering graphs with cycles, it may happen that some of the edges $(v,\cdot)$ lead to vertices the ant has already visited on its way from u to v. Let w be one such vertex. Then (v,w) must be an incorrect edge because otherwise v is no vertex on a shortest path from u to n since all cycles have positive weight. We argue that preventing the ant from traversing incorrect edges to visited vertices cannot increase the probability of taking some incorrect edge.

Let τ_{cor} denote the sum of pheromone on correct edges $(v,\cdot)$ and $\tau_{\text{inc+vis}}$ denote the sum of pheromone on incorrect edges $(v,\cdot)$ leading to a visited vertex. Then the probability of ant a_u choosing some incorrect edge $(v,\cdot)$ is at most

$$\frac{\tau_{\text{inc}} - \tau_{\text{inc+vis}}}{\tau_{\text{cor}} + \tau_{\text{inc}} - \tau_{\text{inc+vis}}} \leqslant \frac{\tau_{\text{inc}}}{\tau_{\text{cor}} + \tau_{\text{inc}}} \leqslant \tau_{\text{inc}}$$

where the second inequality stems from $\tau_{\text{cor}} + \tau_{\text{inc}} \geqslant 1$ due to Lemma 1. □

The argument used in the proof of Corollary 3 breaks down in case cycles with length 0 are allowed. In this case an ant starting at u might have traversed some vertex w before arriving at $v \neq w$. If cycles of length 0 are allowed, we might have that w is on a shortest path from v to n and so (v, w) is a correct edge. This means that the edge might have received a lot of pheromone. When the ant a_u arrives at v after w it cannot choose this edge as this would close a cycle. But excluding the edge with the largest probability mass can significantly bias the probabilities for the remaining edges $(v, \cdot)$ and increase the probability of a_u taking an incorrect edge.

The following theorem gives upper bounds for $\text{MMAS}_{\text{SDSP}}$, each consisting of two additive terms. Intuitively, the first terms cover waiting times until improvements of best-so-far paths are found. The second terms grow with $1/\rho$. They reflect the time to adapt the pheromones after a change of the best-so-far path. This time is called *freezing time* by Neumann, Sudholt, and Witt [32].

Theorem 4. *Consider a directed acyclic graph G with n vertices and arbitrary (possibly negative) weights and let $\Delta := \Delta(G)$ and $\ell := \ell(G)$. If $\tau_{\min} \leqslant 1/(\Delta\ell)$ the expected optimization time of $\text{MMAS}_{\text{SDSP}}$ on G is $O(n/\tau_{\min} + n\log(1/\tau_{\min})/\rho)$. The bound simplifies to $O(n^3 + n\log(n)/\rho)$ for $\tau_{\min} = 1/n^2$ and to $O(n\Delta\ell + n\log(\Delta\ell)/\rho)$ for $\tau_{\min} = 1/(\Delta\ell)$.*

The two special cases $\tau_{\min} = 1/(\Delta\ell)$ and $\tau_{\min} = 1/n^2$ are the most interesting to us as the former yields the best upper bounds under the specified preconditions. The latter bound is oblivious of the graph at hand; it holds even if Δ and ℓ are unknown.

Proof of Theorem 4. We follow the analysis by Attiratanasunthron and Fakcharoenphol [1]. We say that a vertex u is *processed* if a shortest path from u to n has been found and if all incorrect edges leaving u have pheromone $\tau_{\min}$.

We estimate the expected time until a vertex u has been processed, given that all vertices reachable from u on shortest paths from u to n have already been processed. We first consider the expected time until a shortest path from u to n has been found for the first time. We also say that then vertex u has been *optimized*. By Corollary 2 the probability of choosing an edge that belongs to a shortest path from u to n is at least $\tau_{\min}/2$. Such a shortest path is found if the ant does not choose an incorrect edge until n is reached. As all vertices on all shortest paths are processed, all incorrect edges at some vertex v have pheromone $\tau_{\min}$ and the probability of choosing some incorrect edge is at most $\deg(v)\tau_{\min}$ due to Corollary 3. Hence, the probability of choosing an edge on a shortest path is at least $1 - \deg(v)\tau_{\min} \geqslant 1 - 1/\ell$ if $\tau_{\min} \leqslant 1/(\deg(v)\ell)$. As all shortest paths have at most ℓ edges, the probability that no incorrect edge is chosen is at least $(1 - 1/\ell)^{\ell-1} \geqslant 1/e$ with $e = \exp(1)$. Together, the probability of finding a shortest path from u to n is at least $\tau_{\min}/(2e)$.

The expected time until u is optimized is thus at most $2e/\tau_{\min}$. Afterwards, due to the best-so-far rule, a shortest path from u to n is reinforced automatically in each iteration. The precise path may change, but it is guaranteed that only shortest paths are rewarded and hence the pheromone on incorrect edges decreases in every step. As Lemma 2 in [1] already states, $\ln(\tau_{\max}/\tau_{\min})/\rho$ iterations are enough for the vertex to become processed because the simple calculation

$$(1-\rho)^{\ln(\tau_{\max}/\tau_{\min})/\rho} \cdot \tau_{\max} < e^{-\ln(\tau_{\max}/\tau_{\min})} \cdot \tau_{\max} = \tau_{\min}$$

guarantees that the pheromone level in the worst case drops from $\tau_{\max}$ to $\tau_{\min}$. Hence, the expected time until u is processed is bounded by $2e/\tau_{\min} + \ln(\tau_{\max}/\tau_{\min})/\rho$.

Let $v_1, \ldots, v_n$ be a topological ordering of the vertices in V with $v_n = n$. As we are dealing with a directed acyclic graph, all shortest paths from v_i to v_n only use vertices from $\{v_{i+1}, \ldots, v_n\}$. If $v_{i+1}, \ldots, v_n$ have been processed then we can wait for v_i to become processed using the above argumentation. The expected time until all vertices have been processed is bounded by $n2e/\tau_{\min} + n\ln(\tau_{\max}/\tau_{\min})/\rho = O(n/\tau_{\min} + n\log(1/\tau_{\min})/\rho)$. □

Observe that for $\text{MMAS}_{\text{SDSP}}$, once a shortest path from u has been found, the pheromones are continuously "frozen" towards shortest paths from u in the following $F = \ln(\tau_{\max}/\tau_{\min})/\rho$ iterations. The algorithm n-ANT from [1], however, only updates pheromones in case a new best-so-far path is found. This implies that a shortest path from u has to be found several times, in the worst case in F different iterations, in order to freeze the pheromones in the same way. Hence, using the best-so-far rule of MMAS algorithms leads to better performance results. This adds to the comparison of the 1-ANT and MMAS on pseudo-Boolean problems in [32].

3.2. Refined analysis of pheromone adaptation

In the proof of Theorem 4 we have pessimistically assumed that we have to wait for $\ln(\tau_{\max}/\tau_{\min})/\rho$ iterations after optimizing a vertex v, in order to have a good probability of finding shortest paths that contain v. This pessimistic assumption can be relaxed as follows. Instead of waiting $\ln(\tau_{\max}/\tau_{\min})/\rho$ iterations, an average time of $3/\rho$ iterations after optimizing

a new vertex is enough to guarantee that a shortest path can be found with good probability. For simplicity, we assume in the sequel that w.l.o.g. $3/\rho$ is an integer.

Definition 5. Given a weighted directed graph with a target node n and pheromones on all edges, a vertex u is called *well-processed* if for every shortest path $\mathcal{P}$ from u to n the following condition holds. Let $\mathcal{P} = (p_1, \ldots, p_r, n)$ with $u = p_1$, then for every $1 \leqslant i \leqslant r$ vertex p_i has been optimized for at least $3i/\rho$ iterations.

The definition of well-processedness reflects the fact that vertices close to the target (in terms of the number of edges on shortest paths) are typically optimized earlier than vertices whose shortest paths contain many edges. Hence, the former vertices have been optimized for a longer period of time and the pheromones had more time to "freeze" towards the correct edges. Similar arguments have also been used in [32] for the analysis of an ACO algorithm for pseudo-Boolean optimization and by Sudholt and Witt [43] for the analysis of a binary particle swarm optimizer.

Lemma 6. *Consider a weighted graph $G = (V, E, w)$ with target node n such that all cycles have strictly positive weight. Then for every fixed vertex u the following holds. If all successors of u on shortest paths from u to n are well-processed and $\tau_{\min} \leqslant 1/(\Delta(G)\ell(G))$ then the probability that u is optimized within one iteration is at least $\tau_{\min}/(4e)$.*

Once u is optimized, u becomes well-processed after an additional number of $3/\rho$ iterations.

Proof. Let $\ell := \ell(G)$. Consider a vertex v and temporarily ignore the lower pheromone border $\tau_{\min}$. The sum of pheromones on incorrect edges leaving v is always bounded from above by $1 + \deg(v)\tau_{\min} \leqslant 1 + 1/\ell \leqslant 2$ by Lemma 1. If v has been optimized for t iterations then the sum of these pheromones is at most $2(1-\rho)^t$. Considering the lower pheromone border introduces an error of up to $\tau_{\min}$ for each edge, hence with pheromone borders the sum of pheromones on incorrect edges is at most $\deg(v)\tau_{\min} + 2(1-\rho)^t \leqslant 1/\ell + 2(1-\rho)^t$. The probability of choosing some incorrect edge (assuming the ant has traversed a shortest path so far) is at most $1/\ell + 2(1-\rho)^t$ due to Corollary 3. The probability of choosing an edge on a shortest path is thus at least $1 - 1/\ell - 2(1-\rho)^t$.

Recall from the proof of Theorem 4 that an ant chooses the first edge correctly with probability at least $\tau_{\min}/2$. This proves the first claim for $\ell = 1$. We assume $\ell \geqslant 2$ in the following and observe that $1 - 1/\ell - 2(1-\rho)^t \geqslant 1 - 1/\ell - 2e^{-3j}$ for $t = 3j/\rho$ and all $j \in \mathbb{N}$. Given our preconditions on u, the probability for ant a_u of finding a shortest path from u is at least

$$\frac{\tau_{\min}}{2} \cdot \prod_{j=1}^{\ell-1}\left(1 - \frac{1}{\ell} - 2e^{-3j}\right) \geqslant \frac{\tau_{\min}}{2} \cdot \prod_{j=1}^{\ell-1}\left(1 - \frac{1}{\ell} - 4e^{-3j} + \frac{4e^{-3j}}{\ell}\right)$$

$$= \frac{\tau_{\min}}{2} \cdot \prod_{j=1}^{\ell-1}\left(\left(1 - \frac{1}{\ell}\right)\left(1 - 4e^{-3j}\right)\right) \geqslant \frac{\tau_{\min}}{2e} \cdot \prod_{j=1}^{\ell-1}\left(1 - 4e^{-3j}\right).$$

Using $1 - x \geqslant e^{-2x}$ for $0 \leqslant x \leqslant 1/2$, we finally estimate the $\prod$-term as follows:

$$\prod_{j=1}^{\ell-1}\left(1 - 4e^{-3j}\right) \geqslant \prod_{j=1}^{\ell-1}\left(1 - 4e^{-3j}\right) \geqslant \prod_{j=1}^{\ell-1}\exp\left(-8e^{-3j}\right) \geqslant \exp\left(-8\sum_{j=1}^{\infty}e^{-3j}\right) = \exp\left(-\frac{8}{e^3 - 1}\right) \geqslant \frac{1}{2}.$$

This proves the first claim. The second claim follows from the definition of well-processedness as all vertices on shortest paths from u (including u itself) only need to be optimized for at most an additional number of $3/\rho$ iterations. □

Lemma 6 can be used to lower the upper bound given in Theorem 4 from $O(n/\tau_{\min} + n\log(1/\tau_{\min})/\rho)$ to $O(n/\tau_{\min} + n/\rho)$. This is an improvement to the preliminary results from [23]. We take a further step and prove that the latter bound holds for all directed graphs such that all cycles have strictly positive weight.

3.3. An improved upper bound for $\text{MMAS}_{\text{SDSP}}$

In the proof of Theorem 4 we have used the fact that a vertex is optimized efficiently if all successors on all shortest paths have been processed. More precise, there was a fixed ordering $v_1, \ldots, v_n$ of the vertices such that if v_i is the last non-optimized vertex in this ordering, then all shortest paths from v_i only use vertices from $v_{i+1}, \ldots, v_n$. In general directed graphs we cannot argue with a topological ordering. If we require cycles to have strictly positive weight, we can, however, use the maximum number of edges on any shortest path to obtain the desired order.

Theorem 7. *Consider an arbitrary directed graph G with n vertices and strictly positive weights and let $\Delta := \Delta(G)$ and $\ell := \ell(G)$. If $\tau_{\min} \leqslant 1/(\Delta\ell)$ the expected optimization time of $\text{MMAS}_{\text{SDSP}}$ on G is $O(n/\tau_{\min} + n/\rho)$. The bound simplifies to $O(n^3 + n/\rho)$ for $\tau_{\min} = 1/n^2$ and to $O(n\Delta\ell + n/\rho)$ for $\tau_{\min} = 1/(\Delta\ell)$.*

Proof. Let $v_1, v_2, \ldots, v_n$ be an enumeration of all vertices such that $\ell(v_1) \geqslant \ell(v_2) \geqslant \cdots \geqslant \ell(v_n)$ where $\ell(v)$ denotes the maximum number of edges on any shortest path from v to n. Due to the ordering of the vertices, all shortest paths from v_i to v_n only use vertices from $\{v_{i+1}, \ldots, v_n\}$, since otherwise there would exist a shortest path from v_i to n with more than $\ell(v_i)$ edges. Moreover, when following any shortest path, the indices of the vertices are strictly increasing.

We partition a run of the algorithm into phases that are traversed in decreasing order. Phase n starts after the initialization. For $1 \leqslant i \leqslant n$, Phase i ends once all vertices in $\{v_i, \ldots, v_n\}$ are well-processed. After Phase i has ended, Phase $i-1$ starts if $i > 1$. Note that if v_i happens to become well-processed before v_{i+1}, then Phase i will be empty. Also note that the time until Phase 1 has ended is an upper bound for the optimization time. Fix i and assume that Phase $i+1$ has ended. As $\{v_{i+1}, \ldots, v_n\}$ are well-processed, the conditions from Lemma 6 are fulfilled. Hence, the probability of optimizing v_i in one iteration is bounded from below by $\tau_{\min}/(4e)$. The expected waiting time for this event is at most $4e/\tau_{\min}$. By the second claim of Lemma 6 after an additional number of $3/\rho$ steps v_i is well-processed as well and Phase i ends.

The expected time spent in Phase i is bounded by $4e/\tau_{\min} + 3/\rho$. Hence, the expected time until Phase 1 is completed is bounded by $4en/\tau_{\min} + 3n/\rho = O(n/\tau_{\min} + n/\rho)$. □

Up to now, our analyses relied on the vertices being optimized and becoming processed one after another. For certain graphs, however, there may be subgraphs that can be optimized in parallel. Under some additional conditions, we will improve our upper bounds by a factor of ℓ/n. Thereby, we also show that the running time bounds hold with high probability (i.e. with probability at least $1 - n^{-c}$ for some $c > 0$). In the proof we follow ideas from [7] showing that the random time until a short path of length $\ell = \Omega(\log n)$ is found is highly concentrated around the expectation.[2] To prove this formally, we utilize the following Chernoff bound [30].

Lemma 8 *(Chernoff bound). Let X_i, $1 \leqslant i \leqslant t$, be mutually independent random variables and $X := \sum_{i=1}^{t} X_i$. If $\text{Prob}(X_i = 1) = p_i$ and $\text{Prob}(X_i = 0) = 1 - p_i$ where $0 < p_i < 1$, i.e., all X_i are distributed according to a Bernoulli distribution, then for all $0 < \delta \leqslant 1$ the following holds*

$$\text{Prob}\big(X \leqslant (1-\delta)\cdot \text{E}(X)\big) \leqslant \exp\left(-\frac{\delta^2 \cdot \text{E}(X)}{2}\right).$$

The following theorem requires all shortest paths to be unique, which implies that all cycles have strictly positive length.

Theorem 9. *Consider a directed graph G with n vertices where all shortest paths are unique. Let $\Delta := \Delta(G)$, $\ell := \ell(G)$, and $\ell^* := \max\{\ell, \ln n\}$. If $\tau_{\min} \leqslant 1/(\Delta\ell)$ the optimization time of $\text{MMAS}_{\text{SDSP}}$ on G is $O(\ell^*/\tau_{\min} + \ell/\rho)$ with probability at least $1 - 1/n^2$. The bound simplifies to $O(\ell^* n^2 + \ell/\rho)$ for $\tau_{\min} = 1/n^2$ and to $O(\Delta\ell\ell^* + \ell/\rho)$ for $\tau_{\min} = 1/(\Delta\ell)$. The optimization time bound also holds in expectation.*

Proof. Fix a vertex u and the unique shortest path $u = v_{\ell'}, v_{\ell'-1}, \ldots, v_0 = n$ with $\ell' \leqslant \ell$. We pessimistically estimate the expected time until u becomes well-processed, using the phase argumentation from the previous proof in a more formal way. Let T_i be the random time until v_i is optimized. Consider random variables $X_1, \ldots, X_T$ that are independently set to 1 with probability $\tau_{\min}/(4e)$ and to 0 otherwise. The random first point of time T_1^* where $X_{T_1^*} = 1$ stochastically dominates the random time until v_1 is optimized. As v_1 becomes well-processed after an additional waiting time of $3/\rho$ steps, $T_1^* + 3/\rho$ stochastically dominates T_1. Inductively, we have that $T_{\ell'}^* + 3\ell'/\rho$ stochastically dominates $T_{\ell'}$ and hence the time until u is well-processed.

Let $T := 32e\ell^*/\tau_{\min}$ and $X := \sum_{i=1}^{T} X_i$. We have $\text{E}(X) = T \cdot \tau_{\min}/(4e) = 8\ell^*$. By the Chernoff bound from Lemma 8

$$\text{Prob}\big(X < \ell^*\big) \leqslant \text{Prob}\big(X \leqslant (1-7/8)\cdot \text{E}(X)\big) \leqslant e^{-8\ell^*(7/8)^2/2} < e^{-3\ell^*} \leqslant n^{-3}.$$

Hence, the probability that u is not well-processed after $T + 3\ell/\rho$ steps is $1/n^3$. By the union bound, the probability that there is a non-well processed vertex remaining after this time is at most $1/n^2$. The result on the expectation follows from the first result, which holds for arbitrary initial pheromones. If the algorithm does not find all shortest paths within the first $T + 3\ell/\rho$ steps, we repeat the argumentation with another period of this length. The expected number of periods needed is clearly $O(1)$. □

We remark that the condition of all shortest paths being unique can be dropped at the expense of an additional factor $\log n$ in the first term of the running time bound. Partition all vertices into layers where Layer i contains all vertices whose maximum number of edges over all shortest paths to n equals i. The benefit of this layering is that once all vertices in the Layers $i-1, \ldots, 1$ are optimized and well-processed, then all vertices in Layer i can be optimized with probability at

[2] There is a subtle difference to [7]: in their definition of ℓ the authors only consider shortest paths *with a minimum number of edges*, formally $\ell(G, v) := \max_u \min_{p \in \mathcal{P}_{u,v}} \{\#\text{edges on } p\}$ if $\mathcal{P}_{u,v}$ denotes the set of all shortest paths from u to v. Our definition corresponds to setting $\ell(G, v) := \max_u \max_{p \in \mathcal{P}_{u,v}} \{\#\text{edges on } p\}$. Both definitions for ℓ are, however, equal if all shortest paths are unique or have the same number of edges.

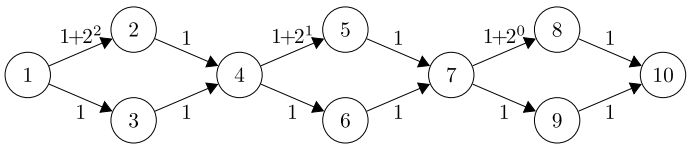

Fig. 2. Example graph for $n = 10$.

least $\tau_{\min}/(4e)$. The expected time until the last vertex in Layer i has been optimized is bounded by $O(\log(n)/\tau_{\min})$ using standard arguments. Summing up these expected waiting times as well as waiting times $3/\rho$ for well-processedness for all layers results in the bound $O(\ell \log(n)/\tau_{\min} + \ell/\rho)$.

All upper bounds on the optimization time so far grow with $\ell(G)$. Our arguments rely on bounding the time required to process the vertices $v_0, \ldots, v_\ell$ one by one. The obvious question arises if this is always necessary. Consider the example graph sketched in Fig. 2 and focus on the vertices with two outgoing edges, called lower edge and upper edge. Every path starting with a lower edge is shorter than every path starting with the respective upper edge. This implies that once an ant takes the lower edge, only the lower edge will be reinforced in the future. The expected time until this happens is bounded by $O(1/\tau_{\min})$ iterations. Hence, after $O(\log(n)/\tau_{\min} + \log(1/\tau_{\min})/\rho)$ iterations all vertices will have pheromone $\tau_{\min}$ on their upper edges, with high probability. In this situation, the probability of finding a shortest path for any ant is at least $1/e$, which adds a term of $O(\log n)$ to the expected time until the last vertex is optimized. The optimization time of $\text{MMAS}_{\text{SDSP}}$ on the example graph is thus $O(\log(n)/\tau_{\min} + \log(1/\tau_{\min})/\rho)$ with high probability. Note that in this case the upper bound $O(n/\tau_{\min} + n/\rho)$ from Theorem 9 is too pessimistic. In general, the bound is, however, tight as will be shown in the sequel.

3.4. *Lower bounds for* $\text{MMAS}_{\text{SDSP}}$

We now turn to lower bounds on the expected optimization time of $\text{MMAS}_{\text{SDSP}}$. The second term in the upper bounds presented so far grows with $1/\rho$. One obvious question is whether this dependence on ρ is really necessary. The following result gives a positive answer, at least unless ρ is so small that the algorithm behaves similar to random search. In this case, we obtain a lower bound that is almost of order $e^{\Omega(\ell)}$.

The main idea for the following theorem is that the pheromones need some time to adapt, such that a shortest path with ℓ edges can be found with good probability. The lower bound applies to all graphs with the following natural property. There is a unique shortest path of ℓ edges. When an ant follows this path, it has to choose between at least two edges at every vertex (except for the last $O(1)$ vertices). This condition ensures that at almost every vertex there is a decision to make between a correct and at least one wrong edge.

Theorem 10. *Consider a weighted graph $G = (V, E)$ with n vertices. Let $\Delta := \Delta(G)$ and $\ell := \ell(G)$. Assume that G contains a unique shortest path $p_0, \ldots, p_\ell = n$ such that for $0 \leqslant i \leqslant \ell - O(1)$ vertex p_i has no edges leading to $\{p_0, \ldots, p_{i-1}\}$ and at least two edges leading to $\{p_{i+1}, \ldots, p_\ell\}$. If $0 < \rho \leqslant 1 - \Omega(1)$ then for every constant $\varepsilon > 0$ the expected optimization time of $\text{MMAS}_{\text{SDSP}}$ on G with $\tau_{\min} \leqslant 1/(\Delta\ell)$ is $\Omega(\min\{(\log \ell)/\rho,\ e^{\ell^{1-\varepsilon}/4}\})$.*

Proof. Consider the ant starting at p_0 trying to create $p_0, \ldots, p_\ell$. By assumption on G we have that while the ant follows this path, at every vertex $0 \leqslant i \leqslant \ell - O(1)$ there are at least two edges to yet unvisited vertices. Let $c > 1$ be a constant, chosen later, which may depend on ε. During the first $t := \min\{(1/\rho - 1) \cdot \varepsilon \ln(\ell),\ e^{\ell^{1-\varepsilon}/4}/c\} = \Omega(\min\{(\log \ell)/\rho,\ e^{\ell^{1-\varepsilon}/4}\})$ steps (using $1/\rho - 1 = \Omega(1/\rho)$ by assumption on ρ) the pheromone on every such edge is at least

$$\frac{1}{\deg(u)} \cdot (1-\rho)^t \geqslant \frac{1}{\deg(u)} \cdot e^{-\varepsilon \ln(\ell)} = \frac{1}{\deg(u)} \cdot \frac{1}{\ell^\varepsilon}.$$

Note that this even holds in case the lower pheromone border is hit. As the probability of taking a specific incorrect edge is at least $p := 1/(2\deg(u)\ell^\varepsilon)$, the probability that the ant takes a correct edge on the path is at most $1 - (\deg(u) - 1) \cdot p = 1 - (\deg(u) - 1) \cdot 1/(2\deg(u)\ell^\varepsilon) \leqslant 1 - 1/(4\ell^\varepsilon)$. The probability that the path $p_0, \ldots, p_\ell$ is created in a specific iteration $t' \leqslant t$ is hence bounded by $(1 - 1/(4\ell^\varepsilon))^{\ell - O(1)} \leqslant O(e^{-\ell^{1-\varepsilon}/4})$. The probability that this happens during the first t iterations is bounded by $t \cdot O(e^{-\ell^{1-\varepsilon}/4}) \leqslant 1/2$ if the constant c in the definition of t is chosen appropriately. Hence with probability at least $1/2$ we have not found all shortest paths after t steps and the lower bound $t/2 = \Omega(\min\{(\log \ell)/\rho,\ e^{\ell^{1-\varepsilon}/4}\})$ follows. □

In order to assess whether the upper bound from Theorem 9 is asymptotically tight, we consider the following class of input instances (see Fig. 3). The basic idea is that the algorithm is forced to optimize the vertices on a chain of $\ell + 1$ vertices one after another, from right to left. The graphs may contain further vertices connected with the destination. This way, the instance class contains graphs of varying ℓ-value, particularly graphs with $\ell < n - 1$.

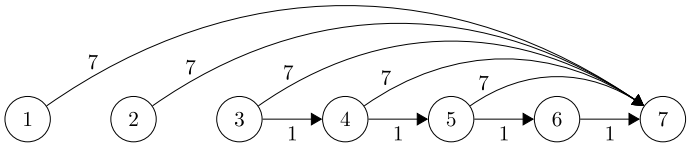

Fig. 3. Example graph $G^{\text{lb}}_{n,\ell}$ from Definition 11 for $n=7$ and $\ell=4$.

Definition 11. Let $G^{\text{lb}}_{n,\ell}=(V,E,w)$, $n\in\mathbb{N}$, $1\leqslant\ell<n$, with $V=\{1,\dots,n\}$, $E=\{(i,i+1)\mid n-\ell\leqslant i\leqslant n-1\}\cup\{(i,n)\mid 1\leqslant i\leqslant n-2\}$, and weights $w((u,v))=1$ if $v=u+1$ and $w((u,v))=n$ if $v\neq u+1$.

Theorem 9 yields an upper bound $O(\ell/\tau_{\min}+\ell/\rho)$ for $G^{\text{lb}}_{n,\ell}$ if $\ell=\Omega(\log n)$ and $\tau_{\min}\leqslant 1/(\Delta(G^{\text{lb}}_{n,\ell})\ell)$. The following lower bound is tight with the upper bound if $\rho=\Omega(\tau_{\min})$. For smaller ρ there is a gap of $O(\log n)$. The proof is inspired by a corresponding lower bound for the function LeadingOnes in pseudo-Boolean optimization [32, Theorem 7].

Theorem 12. *Let $\Delta=\Delta(G^{\text{lb}}_{n,\ell})$. If $1/\text{poly}(n)\leqslant\rho\leqslant 1/2$ and $\ell\geqslant c\log^2 n$ for a large enough constant c then the expected optimization time of* $\text{MMAS}_{\text{SDSP}}$ *on $G^{\text{lb}}_{n,\ell}$ with $1/\text{poly}(n)\leqslant\tau_{\min}\leqslant 1/(\Delta\ell)$ is $\Omega\big(\ell/\tau_{\min}+\frac{\ell}{\rho\log(1/\rho)}\big)$.*

Proof. We ignore the vertices $1,\dots,n-\ell-1$ as finding shortest paths for these vertices is trivial. Consider all paths from u to n with $n-\ell\leqslant u\leqslant n-2$. The path (u,n) has length n. All other paths start with the edge $(u,u+1)$. The length of the path only traversing edges with weight 1 is $n-u$. However, if the path ends with an edge (v,n) for $u<v\leqslant n-2$, the path has length $v-u+n>n$. Hence the path (u,n) is the unique second best path from u to n.

Call a vertex $n-\ell\leqslant u\leqslant n-2$ *wrong* if the best-so-far path found by ant a_u is (u,n). After initialization both edges have an equal probability of being chosen by the first ant. By Chernoff bounds at least $\ell/3$ ants a_u with $n-\ell\leqslant u\leqslant n-2$ choose the incorrect edge with probability $1-e^{-\Omega(\ell)}$ (see Lemma 8) and then the corresponding vertices remain wrong until the shortest path has been found. In the following we assume that we initially have at least $\ell/3$ wrong vertices.

Intuitively, the probability of optimizing a vertex decreases with the number of wrong vertices on its shortest path. The following lemma gives a lower bound on the time until a wrong vertex is optimized if its shortest path contains more than $8\log(1/\rho)$ wrong vertices.

Lemma 13. *Let u be a wrong vertex such that the shortest path from u to n contains $8\log(1/\rho)+i$ wrong vertices, for some $i\in\mathbb{N}$. Then the probability that u is optimized within the next $1/\rho-1$ steps is at most $\big(1-\frac{1}{4e}\big)^i$.*

Proof. Let W be the set of wrong vertices on the shortest path from u to n. As long as a vertex remains wrong, the pheromone on its incorrect edge is at least $1/2$. (It even increases continuously towards $\tau_{\max}$ unless the shortest path is found.) This means for every vertex in W the current probability of choosing the incorrect edge is at least $1/4$ using Corollary 2. Once a vertex in W becomes optimized, the probability decreases by a factor of $(1-\rho)$ in each following iteration. Let $t:=1/\rho-1$. Using the factor $(1-\rho)^t$ for a crude estimate yields that during the next t iterations the probability of choosing the incorrect edge is always at least $1/4\cdot(1-\rho)^t\geqslant 1/(4e)$. This implies that the probability of finding the shortest path from u to n is at most $(1-1/(4e))^{8\log(1/\rho)+i}$. Taking the union bound for t iterations, the probability that u is optimized within the next t iterations is at most

$$t\left(1-\frac{1}{4e}\right)^{8\log(1/\rho)+i}\leqslant 2^{\log(1/\rho)}\left(1-\frac{1}{4e}\right)^{8\log(1/\rho)+i}\leqslant\left(1-\frac{1}{4e}\right)^{i}.\qquad\square$$

The following lemma is an immediate conclusion.

Lemma 14. *The expected number of wrong vertices which are optimized within $1/\rho-1$ iterations is bounded by $8\log(1/\rho)+4e$.*

Proof. Let W be the set of all wrong vertices and let p_u denote the probability that a wrong vertex u becomes optimized within $1/\rho-1$ iterations. If u has at most $8\log(1/\rho)$ wrong vertices on its shortest path, we estimate $p_u\leqslant 1$. Otherwise, the shortest path contains $8\log(1/\rho)+i$ wrong vertices and Lemma 13 yields $p_u\leqslant(1-1/(4e))^i$. The expected number of optimized vertices is hence at most

$$\sum_{u\in W}p_u\leqslant 8\log(1/\rho)+\sum_{i=0}^{\infty}\left(1-\frac{1}{4e}\right)^i=8\log(1/\rho)+4e.\qquad\square$$

Recall that we assume at least $\ell/3$ wrong vertices in the beginning. Now consider $\ell/6\cdot 1/(8\log(1/\rho)+4e)$ subsequent periods of $1/\rho-1$ iterations each. Note that this is $\Omega(\ell/(\rho\log(1/\rho)))$ iterations in total. Lemma 14 implies that the

expected number of vertices optimized within all periods is bounded by $\ell/6$. By Markov's inequality, the probability that $\ell/3$ vertices are corrected is at most $1/2$. Taking into account the probability of not having at least $\ell/3$ wrong vertices in the beginning, we have that with probability at least $1/2 - e^{-\Omega(\ell)} = \Omega(1)$ the algorithm needs at least $\Omega(\ell/(\rho \log(1/\rho)))$ iterations. This establishes the lower bound $\Omega(\ell/(\rho \log(1/\rho)))$ on the expected optimization time.

It remains to show a lower bound $\Omega(\ell/\tau_{\min})$ to complete the theorem. By Lemma 14 after $F := \ln(\tau_{\max}/\tau_{\min})/\rho$ iterations the expected number of wrong vertices that have been optimized is $O(\ln(\tau_{\max}/\tau_{\min}) \cdot \log(1/\rho)) = O(\log^2 n)$ due to our assumptions on $\tau_{\min}$ and ρ. With probability at least $1/2$ this random number is at most $O(\log^2 n)$. Assuming this to happen, the number of wrong vertices left after F iterations is $\ell/3 - O(\log^2 n) = \Omega(\ell)$ if the constant c in the prerequisite $\ell \geqslant c \log^2 n$ is chosen large enough. For these wrong vertices, the pheromones have been frozen towards their incorrect edges.

Now the probability of optimizing some vertex v if its shortest path contains at least 4 wrong vertices is at most $(\tau_{\min})^4 \leqslant \tau_{\min}/\ell^3$. Hence, with probability at least $1 - O(1/\ell)$ during $\Omega(\ell/\tau_{\min})$ iterations it does not happen that more than 4 wrong vertices are corrected in the same iteration. The probability of correcting a wrong vertex with $i-1$ wrong successors on its shortest path is at most $\tau_{\min}^i$. Hence, the probability that some wrong vertex is corrected is at most $\sum_{i=1}^{\infty} \tau_{\min}^i = O(\tau_{\min})$ and the expected time until some wrong vertex is corrected is $\Omega(1/\tau_{\min})$. If the number of wrong vertices always decreases by at most 4, $\Omega(\ell)/4 = \Omega(\ell)$ of these events need to happen before all shortest paths are found. The total expected time in this setting is then at least $\Omega(\ell/\tau_{\min})$. As the described setting occurs with probability at least $1/2 - O(1/\ell) - e^{-\Omega(\ell)} = \Omega(1)$, we obtain an unconditional bound of the same order. □

Note that the lower bound only holds for $\tau_{\min} \leqslant 1/(\Delta\ell)$ and the first term of the lower bound grows with $1/\tau_{\min}$. The reader might be tempted to choose $\tau_{\min} > 1/(\Delta\ell)$ to achieve an expected optimization time of $o(\Delta\ell^2)$. However, if $\tau_{\min}$ is too large, the pheromones cannot adapt properly and the search is too close to random search. Consider, for instance, the vertex $u = n - \ell$ in $G^{\text{lb}}_{n,\ell}$ whose shortest path contains ℓ edges. Even if all pheromones are best-possible, the probability of choosing a correct edge at some vertex $v \leqslant n-2$ is still at most $1 - \tau_{\min}/2$. The probability of making the right decision $\ell - 1$ times is then only $(1 - \tau_{\min}/2)^{\ell-1} \approx e^{-\tau_{\min}\ell/2}$. The reciprocal of this probability constitutes a lower bound on the expected optimization time; it is superpolynomial if $\tau_{\min} = \omega((\log n)/\ell)$. This also justifies why our upper bounds so far are restricted to $\tau_{\min} \leqslant 1/(\Delta\ell)$.

3.5. An adaptive choice of pheromone borders

One exception from the rule might be graphs where only few vertices have high degree. Recall that the probability of constructing a shortest path from u, given that all successors of u on shortest paths are well-processed, is bounded below by $\tau_{\min}/(4e)$ if $\tau_{\min} \leqslant 1/(\deg(u)\ell)$ (see Lemma 6). If the same pheromone border applies to all edges, the best choice for $\tau_{\min}$ where this condition holds for all vertices is $\tau_{\min} = 1/(\Delta\ell)$.

However, we might as well consider an ACO system where pheromone borders can be adapted to single vertices. The pheromone on an edge $e = (u, \cdot)$ is then bounded by the pheromone borders $\tau_{\min}(u)$ and $\tau_{\max}(u)$. If $\tau_{\min}(u) = 1/(\deg(u)\ell)$ and $\tau_{\max}(u) = 1 - \tau_{\min}(u)$ then the expected waiting time until u is optimized, given that all successors on shortest paths are well-processed, is bounded by $4e/\tau_{\min}(u) = 4e \deg(u)\ell$. Summing up these waiting times for all vertices and adding waiting times until the vertices become well-processed leads to the following bound.

Theorem 15. *Consider a weighted graph G with n vertices and m edges such that all cycles have strictly positive weight. Let $\ell := \ell(G)$. The expected optimization time of* $\text{MMAS}_{\text{SDSP}}$ *using adaptive pheromone borders with $\tau_{\min}(u) = 1/(\deg(u)\ell)$ for all vertices u is $O(\ell m + n/\rho)$.*

4. All-pairs shortest path problem

We now extend $\text{MMAS}_{\text{SDSP}}$ towards an algorithm $\text{MMAS}_{\text{APSP}}$ for the APSP. For each destination $v \in V$ we introduce a distinct pheromone function $\tau_v : E \to \mathbb{R}_0^+$. In each iteration, on each vertex u, and for each destination v we have an ant $a_{u,v}$ starting at u and heading for v. An ant heading for v uses the pheromone function τ_v for orientation and it updates τ_v as described in Section 2. $\text{MMAS}_{\text{APSP}}$ remembers the best-so-far paths $p^*_{u,v}$ from u to v for all $u, v \in V$.

The following result is an immediate implication from Theorem 9.

Theorem 16. *Consider a directed graph G with n vertices where all shortest paths are unique. Let $\Delta := \Delta(G)$, $\ell := \ell(G)$, and $\ell^* := \max\{\ell, \ln n\}$. If $\tau_{\min} \leqslant 1/(\Delta\ell)$ the optimization time of* $\text{MMAS}_{\text{APSP}}$ *on G is $O(\ell^*/\tau_{\min} + \ell/\rho)$ with probability at least $1 - 1/n$. The bound simplifies to $O(\ell^* n^2 + \ell/\rho)$ for $\tau_{\min} = 1/n^2$ and to $O(\Delta\ell\ell^* + \ell/\rho)$ for $\tau_{\min} = 1/(\Delta\ell)$. The optimization time bound also holds in expectation.*

Proof. Fix a destination v. Due to Theorem 9, within $O(\ell^*/\tau_{\min} + \ell/\rho)$ iterations shortest paths from all sources u to the destination v are found with probability at least $1 - 1/n^2$. Hence, using the union bound all shortest paths are found with probability at least $1 - 1/n$. □

Algorithm 3 Path construction from u to v for $\text{MMAS}_{\text{APSP}}$ with interaction

```
Choose p ∈ [0, 1) uniformly at random
if p < p_int then
    Choose w ∈ V uniformly at random
    Construct a simple path p' = (p'_0, ..., p'_{ℓ'}) from u to w w.r.t. τ_w
    Construct a simple path p'' = (p''_0, ..., p''_{ℓ''}) from w to v w.r.t. τ_v
    if p'_{ℓ'} = w then p ← (p'_0, ..., p'_{ℓ'}, p''_1, ..., p''_{ℓ''}) else p ← p' end if
else
    Construct a simple path p from u to v w.r.t. τ_v
end if
return p
```

We see that ants heading for different destinations do not collaborate in our ant system since ants heading for a destination v concern for the pheromone function τ_v exclusively. Therefore we could also run n instances of $\text{MMAS}_{\text{SDSP}}$ in parallel to achieve the same result. An obvious question is whether the ants can interact in some clever way to achieve a better result.

Interestingly, the following simple mechanism proves useful. Consider the ant $a_{u,v}$ heading for vertex v. Instead of always using the pheromone function τ_v to travel to v, with some probability p_{int} the ant decides to follow foreign pheromones. It first chooses an intermediate destination w uniformly at random, then uses the pheromone function τ_w to travel to w, and afterwards uses the pheromone function τ_v to travel to the final destination v (see Algorithm 3). The pheromone update for ant $a_{u,v}$ always applies to the pheromones τ_v as before.'

Note that the proposed interaction mechanism resembles crossover operators used in genetic algorithms (see [8,12]). The interaction mechanism allows to combine a shortest path with another shortest path yielding a longer shortest path in a similar fashion as the mentioned crossover operators do. With this mechanism the ant $a_{u,v}$ can profit from useful information laid down by other ants that headed towards w, in particular if w happens to be a vertex on a shortest path from u to v. The following theorem gives a significantly improved bound, without restriction to graphs with unique shortest paths.

Theorem 17. *Consider a weighted graph G with n vertices such that all cycles have strictly positive weight. Let $\Delta := \Delta(G)$ and $\ell := \ell(G)$. If $\rho \leqslant \rho_{\text{ub}} := (1 - p_{\text{int}})/(12\Delta \log n)$ then the optimization time of $\text{MMAS}_{\text{APSP}}$ using interaction (with some constant probability $0 < p_{\text{int}} < 1$) on G with $1/n^3 \leqslant \tau_{\min} \leqslant 1/(\Delta\ell)$ is $O(n \log n + \log(\ell) \log(1/\tau_{\min})/\rho)$ with probability at least $1 - 1/n^2$. The bound simplifies to $O(n \log n + \log(\ell) \log(n)/\rho)$ for $\tau_{\min} = 1/n^2$ and to $O(n \log n + \log(\ell) \log(\Delta\ell)/\rho)$ for $\tau_{\min} = 1/(\Delta\ell)$. The optimization time bound also holds in expectation.*

Proof. We introduce similar notions as before. Consider a pair (u, v) of vertices. Let $\ell_{u,v} := \max\{\#\text{edges on } p \mid p \text{ is a shortest path from } u \text{ to } v\}$. We call an edge *incorrect* with respect to v if it does not belong to a shortest path to v. We call (u, v) *optimized* if a shortest path from u to v has been found. We call (u, v) *processed* if it has been optimized and if the pheromone $\tau_v(\cdot)$ on all edges $(u, \cdot)$, which are incorrect with respect to v, is $\tau_{\min}$.

Consider the first $t = \lceil (\ln 2)/\rho_{\text{ub}} \rceil = O(\Delta \log n)$ iterations. Consider a pair (u, v) with $\ell_{u,v} = 1$. The probability of optimizing (u, v) in iteration i is at least $(1 - p_{\text{int}}) \cdot (1 - \rho)^{i-1}/(2\Delta) \geqslant (1 - p_{\text{int}}) \cdot (1 - \rho_{\text{ub}})^{i-1}/(2\Delta)$ since the ant $a_{u,v}$ decides with probability $1 - p_{\text{int}}$ to head for v and chooses (u, v) with probability at least $(1 - \rho)^{i-1}/(2\Delta)$ due to Corollary 2. Hence, the probability of *not* optimizing (u, v) within the considered phase is at most

$$\begin{aligned}\prod_{i=1}^{t}\left(1 - \frac{(1 - p_{\text{int}}) \cdot (1 - \rho_{\text{ub}})^{i-1}}{2\Delta}\right) &\leqslant \exp\left(-\frac{1 - p_{\text{int}}}{2\Delta} \cdot \sum_{i=0}^{t-1}(1 - \rho_{\text{ub}})^{i}\right) \\ &= \exp\left(-\frac{1 - p_{\text{int}}}{2\Delta} \cdot \frac{1 - (1 - \rho_{\text{ub}})^{t}}{1 - (1 - \rho_{\text{ub}})}\right) \\ &\leqslant \exp\left(-\frac{1 - p_{\text{int}}}{4\Delta\rho_{\text{ub}}}\right).\end{aligned}$$

Since $\rho_{\text{ub}} = (1 - p_{\text{int}})/(12\Delta \log n) \leqslant (1 - p_{\text{int}})/(4\Delta \ln(2n^4))$, the above probability is at most $1/(2n^4)$. Because of the union bound, all pairs (u, v) with $\ell_{u,v} = 1$ are optimized within the considered phase with probability at least $1 - f_1$ where $f_1 := 1/(2n^2)$. We already know that an optimized pair (u, v) is processed within $\lceil \ln(\tau_{\max}/\tau_{\min})/\rho \rceil$ iterations.

Consider a pair (u, v) of vertices and fix a shortest path $p_{u,v} = (v_0, \ldots, v_{\ell_{u,v}})$ from u to v with $\ell_{u,v}$ edges. Let i with $(3/2)^i < \ell_{u,v} \leqslant (3/2)^{i+1}$. We derive a lower bound for the probability of optimizing (u, v) assuming that all pairs (u', v') with $\ell_{u',v'} \leqslant (3/2)^i$ are processed. The ant decides to choose an intermediate destination w with probability p_{int}, which is situated on the middle third $(v_{\ell_{u,v} - \lfloor (3/2)^i \rfloor}, \ldots, v_{0 + \lfloor (3/2)^i \rfloor})$ of the path $p_{u,v}$ with probability

$$\frac{\lfloor (3/2)^i \rfloor - \ell_{u,v} + \lfloor (3/2)^i \rfloor + 1}{n} = \frac{2 \cdot \lfloor (3/2)^i \rfloor - \ell_{u,v} + 1}{n} \geqslant \frac{\ell_{u,v}}{3n}.$$

Hence, the number of edges of all shortest paths $p_{u,w}$ ($p_{w,v}$) from u (w) to w (v) is at most $(3/2)^i$. Since (x,w) ((x,v)) is processed for all vertices x on a shortest path from u (w) to w (v), the ant follows a shortest path from u to v with probability at least $(1-1/\ell)^{\ell-1} \geqslant 1/e$. Altogether, we have shown that the probability of optimizing (u,v) is at least $p_{\text{int}} \cdot \ell_{u,v}/(3n) \cdot 1/e > p_{\text{int}} \cdot (3/2)^i/(3en)$.

We divide a run of the ant system into phases. The ith phase finishes with all pairs (u,v) with $(3/2)^{i-1} < \ell_{u,v} \leqslant (3/2)^i$ being processed. Since $\ell_{u,v} \leqslant \ell$, we have to consider $\alpha := \lceil \log(\ell)/\log(3/2) \rceil$ phases.

Consider Phase i of length $t = \lceil 3en/(p_{\text{int}} \cdot (3/2)^i) \cdot \ln(2\alpha n^4) \rceil$. The probability of *not* optimizing a pair (u,v) with $(3/2)^{i-1} < \ell_{u,v} \leqslant (3/2)^i$ within the phase is at most $(1 - p_{\text{int}} \cdot (3/2)^i/(3en))^t \leqslant 1/(2\alpha n^4)$. Due to the union bound, all such pairs (u,v) are optimized within t iterations with probability at least $1 - 1/(2\alpha n^2)$. We know that an optimized pair (u,v) is processed within $\lceil \ln(\tau_{\max}/\tau_{\min})/\rho \rceil$ iterations. Using the union bound once again, we conclude that all phases are finished within

$$\sum_{i=1}^{\alpha} \left(\left\lceil \frac{3en \cdot \ln(2\alpha n^4)}{p_{\text{int}} \cdot (3/2)^i} \right\rceil + \left\lceil \frac{\ln(\tau_{\max}/\tau_{\min})}{\rho} \right\rceil \right) \leqslant \frac{3en \cdot \ln(2\alpha n^4)}{p_{\text{int}}} \cdot \sum_{i=1}^{\alpha} \left(\frac{2}{3}\right)^i + \alpha \cdot \frac{\ln(1/\tau_{\min})}{\rho} + 2\alpha$$
$$= O\left(n \log n + \log(\ell) \log(1/\tau_{\min})/\rho\right)$$

iterations with probability at least $1 - f_2$ where $f_2 := 1/(2n^2)$. The first part of the theorem follows since both failure probabilities f_1 and f_2 sum up to $1/n^2$.

We still have to provide an estimation for the expected optimization time in case of a failure. We know that a pair (u,v) with $\ell_{u,v} = 1$ is optimized independent from the current pheromones with probability $\Omega(\tau_{\min})$. Hence, after an expected number of $O((\log n)/\tau_{\min})$ iterations all such pairs have been optimized. Since the argumentation from the previous paragraphs holds for all initial pheromone values, we obtain the bound $O((\log n)/\tau_{\min} + n \log n + \log(\ell) \log(1/\tau_{\min})/\rho)$ on the expected optimization time. Together with the first bound holding with probability $1 - 1/n^2$ the second part of the theorem follows because the assumption $\tau_{\min} \geqslant 1/n^3$ guarantees $O(1/n^2 \cdot (\log n)/\tau_{\min}) = O(n \log n)$. □

Consider the parameters $\tau_{\min} = 1/(\Delta\ell)$, $\tau_{\max} = 1 - 1/(\Delta\ell)$, and $\rho = (1 - p_{\text{int}})/(12\Delta \log n)$. Observe that adding a new vertex with no incoming edges to a graph cannot decrease the optimization time. If we consider the graph $G^{\text{lb}}_{n,n-1}$ from Section 3 and add a dummy node with out-degree $\Omega(n)$ and no incoming edges, we obtain a graph with linear maximum degree where the lower bound $\Omega(\ell/\tau_{\min}) = \Omega(n^3)$ holds for $\text{MMAS}_{\text{SDSP}}$ (cf. Theorem 12). This lower bound trivially also holds for $\text{MMAS}_{\text{APSP}}$ without interaction. If $\Delta, \ell = \Omega(n)$, the upper bounds given in Theorems 16 and 17 simplify to $O(n^3)$ and $O(n \log^3 n)$, respectively. Hence, the ant system clearly profits from our simple interaction mechanism and more collaboration between the ants. Note that when comparing the numbers of path length evaluations, if $\Delta = O(n/\log^2 n)$ then with an optimal choice of ρ Theorem 17 gives an upper bound of $O(n^3 \log n)$ evaluations. This matches the upper bound for the GA with feasible parent selection from [10].

For larger Δ the same result can be achieved when the evaporation factor ρ is dynamically increased over time. This resembles the concept of simulated annealing where the system becomes more greedy over time. Assume that ρ is adapted such that during the first $\lceil (\ln 2)/\rho_{\text{ub}} \rceil \leqslant cn \log n$ iterations, for some proper constant c, we have $\rho \leqslant (1 - p_{\text{int}})/(12\Delta \log n)$, and after $c'n \log n$ iterations for some constant $c' > c$ we have $\rho = \Omega((\log n)/\Delta)$. In this case the main arguments used in the proof of Theorem 17 go through and the expected optimization time of the system is $O(n \log n + \log(\ell) \log(\Delta\ell)\Delta/(\log n)) = O(n \log n)$.

5. Conclusions and future work

ACO is motivated by the ability of real ant colonies to find shortest paths to a food source. Building on an initial study by Attiratanasunthron and Fakcharoenphol [1], we have conducted a rigorous analysis of the running time of ACO algorithms for shortest path problems. Our results (see Table 1) significantly improve and generalize the previous results for single-destination shortest paths. Taking the number of function evaluations as performance measure, the bound for $\text{MMAS}_{\text{SDSP}}$ is better than the bound for the evolutionary algorithm $(1+1)$ EA [7] with a vertex-based mutation operator if $\Delta\ell = o(n)$ and ρ is not too small. Note, however, that our upper bound for $\text{MMAS}_{\text{SDSP}}$ is not better than the upper bound for the $(1+1)$ EA with an edge-based mutation operator [9].

For all-pairs shortest paths first results have been obtained using $\text{MMAS}_{\text{APSP}}$ as a direct generalization of $\text{MMAS}_{\text{SDSP}}$. We have proved that, surprisingly, letting ants temporarily follow foreign pheromone traces to random destinations yields drastically improved results. This is also the first result for combinatorial optimization where a slow adaptation of pheromones is crucial, i.e., low values for the evaporation factor ρ yield the best upper bounds. For an optimal static choice of ρ when $\Delta = O(n/\log^2 n)$ the bound of $O(n^3 \log n)$ function evaluations matches the best known bound for genetic algorithms [10]. This puts ACO among the currently best known metaheuristics for the all-pairs shortest path problem from a theoretical perspective.

Our upper bounds generalize to various other distance or weight functions. The only requirement to the weight function exploited in our analyses (apart from the special case of infinite weights) is the following condition for all vertices u, w, v. If w is a vertex on a shortest path from u to v then concatenating any shortest path from u to w with any shortest path

from w to v results in a shortest path from u to v. For instance, our results transfer to the computation of the smallest edge weight on paths to the destination.

As newly constructed paths are always evaluated from scratch, our ACO algorithms also work in dynamic environments where the network or weights on the edges are subject to probabilistic changes. An interesting topic for future work is to examine under which conditions the ants are able to efficiently solve dynamic shortest path problems. This also reflects the fact that ACO algorithms are often applied in dynamic settings. A first step in this direction has already been made [24]. Moreover, ACO algorithms may be analyzed on constrained shortest path problems or variants such as routing problems. We expect that the used techniques and the insights gained throughout this work will also prove useful to obtain further results for graph problems like spanning tree problems or the TSP.

Acknowledgements

Dirk Sudholt was partly supported by EPSRC grant EP/D052785/1 and a postdoctoral fellowship from the German Academic Exchange Service while visiting the International Computer Science Institute in Berkeley, CA, USA.

References

[1] N. Attiratanasunthron, J. Fakcharoenphol, A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs, Information Processing Letters 105 (3) (2008) 88–92.
[2] H. Bast, S. Funke, P. Sanders, D. Schultes, Fast routing in road networks with transit nodes, Science 316 (5824) (2007) 566.
[3] S. Baswana, S. Biswas, B. Doerr, T. Friedrich, P.P. Kurur, F. Neumann, Computing single source shortest paths using single-objective fitness functions, in: Proceedings of the 10th International Workshop on Foundations of Genetic Algorithms (FOGA '09), ACM Press, 2009, pp. 59–66.
[4] T.M. Chan, More algorithms for all-pairs shortest paths in weighted graphs, in: Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC '07), ACM Press, 2007, pp. 590–598.
[5] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd edn., MIT Press, 2001.
[6] G. Di Caro, M. Dorigo, AntNet: Distributed stigmergetic control for communications networks, Journal of Artificial Intelligence Research 9 (1998) 317–365.
[7] B. Doerr, E. Happ, C. Klein, A tight analysis of the $(1+1)$-EA for the single source shortest path problem, in: Proceedings of the 8th IEEE Congress on Evolutionary Computation (CEC '07), IEEE Press, 2007, pp. 1890–1895.
[8] B. Doerr, E. Happ, C. Klein, Crossover can provably be useful in evolutionary computation, in: Proceedings of the 10th Genetic and Evolutionary Computation Conference (GECCO '08), ACM Press, 2008, pp. 539–546.
[9] B. Doerr, D. Johannsen, Edge-based representation beats vertex-based representation in shortest path problems, in: Proceedings of the 12th Genetic and Evolutionary Computation Conference (GECCO '10), ACM Press, 2010, pp. 759–766.
[10] B. Doerr, D. Johannsen, T. Kötzing, F. Neumann, M. Theile, More effective crossover operators for the all-pairs shortest path problem, in: Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN '10), in: LNCS, vol. 6238, Springer, 2010, pp. 184–193.
[11] B. Doerr, F. Neumann, D. Sudholt, C. Witt, Runtime analysis of the 1-ANT Ant Colony Optimizer, Theoretical Computer Science 412 (17) (2011) 1629–1644.
[12] B. Doerr, M. Theile, Improved analysis methods for crossover-based algorithms, in: Proceedings of the 11th Genetic and Evolutionary Computation Conference (GECCO '09), ACM Press, 2009, pp. 247–254.
[13] M. Dorigo, C. Blum, Ant colony optimization theory: A survey, Theoretical Computer Science 344 (2005) 243–278.
[14] M. Dorigo, L.M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.
[15] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: An autocatalytic optimizing process, Tech. Rep. 91-016 Revised, Politecnico di Milano, 1991.
[16] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, 2004.
[17] S. Droste, T. Jansen, I. Wegener, Upper and lower bounds for randomized search heuristics in black-box optimization, Theory of Computing Systems 39 (4) (2006) 525–544.
[18] S. Fischer, I. Wegener, The one-dimensional Ising model: Mutation versus recombination, Theoretical Computer Science 344 (2–3) (2005) 208–225.
[19] W.J. Gutjahr, A generalized convergence result for the graph-based ant system metaheuristic, Probability in the Engineering and Informational Sciences 17 (2003) 545–569.
[20] W.J. Gutjahr, Mathematical runtime analysis of ACO algorithms: Survey on an emerging issue, Swarm Intelligence 1 (2007) 59–79.
[21] W.J. Gutjahr, First steps to the runtime complexity analysis of ant colony optimization, Computers and Operations Research 35 (9) (2008) 2711–2727.
[22] W.J. Gutjahr, G. Sebastiani, Runtime analysis of ant colony optimization with best-so-far reinforcement, Methodology and Computing in Applied Probability 10 (2008) 409–433.
[23] C. Horoba, D. Sudholt, Running time analysis of ACO systems for shortest path problems, in: Proceedings of the 2nd International Workshop on Engineering Stochastic Local Search Algorithms (SLS '09), in: LNCS, vol. 5752, Springer, 2009, pp. 76–91.
[24] C. Horoba, D. Sudholt, Ant colony optimization for stochastic shortest path problems, in: Proceedings of the 12th Genetic and Evolutionary Computation Conference (GECCO '10), 2010, pp. 1465–1472.
[25] T. Jansen, I. Wegener, On the analysis of evolutionary algorithms—A proof that crossover really can help, Algorithmica 34 (1) (2002) 47–66.
[26] D.R. Karger, D. Koller, S.J. Phillips, Finding the hidden path: Time bounds for all-pairs shortest paths, SIAM Journal of Computing 22 (1993) 1199–1217.
[27] T. Kötzing, F. Neumann, H. Röglin, C. Witt, Theoretical properties of two ACO approaches for the traveling salesman problem, in: Seventh International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '10), in: LNCS, vol. 6234, Springer, 2010, pp. 324–335.
[28] T. Kötzing, F. Neumann, D. Sudholt, M. Wagner, Simple Max–Min Ant Systems and the optimization of linear pseudo-Boolean functions, in: Proceedings of the 11th Workshop on Foundations of Genetic Algorithms (FOGA 2011), ACM Press, 2011, pp. 209–218.
[29] D. Merkle, M. Middendorf, Modelling the dynamics of ant colony optimization algorithms, Evolutionary Computation 10 (3) (2002) 235–262.
[30] M. Mitzenmacher, E. Upfal, Probability and Computing, Cambridge University Press, 2005.
[31] F. Neumann, D. Sudholt, C. Witt, Rigorous analyses for the combination of ant colony optimization and local search, in: Proceedings of the 6th International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '08), in: LNCS, vol. 5217, Springer, 2008, pp. 132–143.
[32] F. Neumann, D. Sudholt, C. Witt, Analysis of different MMAS ACO algorithms on unimodal functions and plateaus, Swarm Intelligence 3 (1) (2009) 35–68.
[33] F. Neumann, D. Sudholt, C. Witt, A few ants are enough: ACO with iteration-best update, in: Proceedings of the 12th Genetic and Evolutionary Computation Conference (GECCO '10), 2010, pp. 63–70.

[34] F. Neumann, M. Theile, How Crossover speeds up evolutionary algorithms for the multi-criteria all-pairs-shortest-path problem, in: Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN '10), in: LNCS, vol. 6238, Springer, 2010, pp. 667–676.
[35] F. Neumann, C. Witt, Runtime analysis of a simple ant colony optimization algorithm, in: Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC '06), in: LNCS, vol. 4288, Springer, 2006, pp. 618–627.
[36] F. Neumann, C. Witt, Ant colony optimization and the minimum spanning tree problem, in: Proceedings of the 2nd International Conference on Learning and Intelligent Optimization (LION '07), in: LNCS, vol. 5313, Springer, 2008, pp. 153–166.
[37] F. Neumann, C. Witt, Runtime analysis of a simple ant colony optimization algorithm, Algorithmica 54 (2) (2009) 243–255.
[38] J.B. Orlin, K. Madduri, K. Subramani, M. Williamson, A faster algorithm for the single source shortest path problem with few distinct positive lengths, Journal of Discrete Algorithms 8 (2) (2010) 189–198.
[39] J. Scharnow, K. Tinnefeld, I. Wegener, The analysis of evolutionary algorithms on sorting and shortest paths problems, Journal of Mathematical Modelling and Algorithms 3 (4) (2004) 349–366.
[40] T. Stützle, H.H. Hoos, MAX–MIN ant system, Journal of Future Generation Computer Systems 16 (2000) 889–914.
[41] D. Sudholt, Crossover is provably essential for the Ising model on trees, in: Proceedings of the 7th Genetic and Evolutionary Computation Conference (GECCO '05), ACM Press, 2005, pp. 1161–1167.
[42] D. Sudholt, Using Markov-Chain mixing time estimates for the analysis of ant colony optimization, in: Proceedings of the 11th Workshop on Foundations of Genetic Algorithms (FOGA 2011), ACM Press, 2011, pp. 139–150.
[43] D. Sudholt, C. Witt, Runtime analysis of a binary particle swarm optimizer, Theoretical Computer Science 411 (21) (2010) 2084–2100.
[44] V. Vassilevska, Nondecreasing paths in a weighted graph or: How to optimally read a train schedule, in: Proceedings of the 19th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '08), SIAM, 2008, pp. 465–472.
[45] I. Wegener, Towards a theory of randomized search heuristics, in: Mathematical Foundations of Computer Science, in: LNCS, vol. 2747, Springer, 2003, pp. 125–141.
[46] Y. Zhou, Runtime analysis of an ant colony optimization algorithm for TSP instances, IEEE Transactions on Evolutionary Computation 13 (5) (2009) 1083–1092.